Procedia Computer Science

Volume 51, 2015, Pages 2883–2887

ICCS 2015 International Conference On Computational Science

Fully automatic 2D *hp*-adaptive Finite Element Method for Non-Stationary Heat Transfer

Paweł J. Matuszyk, Marcin Sieniek, and Maciej Paszyński

[1] Baker Hughes Inc., Houston Technology Center, Houston, TX, USA

[2] Department of Computer Science, AGH University of Science and Technology, Kraków, Poland

Abstract

In this paper we present a fully automatic *hp* adaptive finite element method code for non-stationary two dimensional problems. The code utilizes the α-scheme for time discretization and fully automatic *hp* adaptive finite element method discretization for numerical solution of each time step. The code is verified on the examplary non-stationary problem of heat transfer over the L-shape domain.

Acknowledgements: The work of MP is supported by Polish National Science Center grant no. 2012/07/B/ST6/01229. The work of MS is supported by a Dean's grant.

Keywords: finite element method, non-stationary problems, fully automatic, *hp* adaptivity

1 Motivation

The the self-adaptive *hp* finite element method code (*hp*-FEM) has been developed by the group of prof. Demkowicz for stationary elliptic and Maxwell problems in two and three dimensions [6, 7]. The code generates a sequence of computational grids delivering exponential convergence of the numerical error with respect to the mesh size, as predicted by theory [4, 5]. In this paper we present an extension of the code for parabolic problems. We test our code on the exemplary non-stationary heat transfer problem over the L-shape domain. This code is a prototype for transient problems.

2 Modelling a non-stationary parabolic problem

In this section we present a model parabolic problem solved by our non-stationary adaptive *hp*-FEM.

Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2015

doi:10.1016/j.procs.2015.05.452

2.1 Strong form of the parabolic problem

The strong form of our model parabolic problem is given by

$$\begin{cases} \rho c_p \dot{u} - \nabla \cdot (k \nabla u) = f & \text{in } \Omega \times I \\ u = u_D(t) & \text{on } \Gamma_D \times I \\ k\, \mathbf{n} \cdot \nabla u = \beta\,(u_N(t) - u) + q(t) & \text{on } \Gamma_N \times I \\ u(\mathbf{x}, 0) = u_0(\mathbf{x}) & \text{in } \Omega \end{cases}, \tag{1}$$

where u is the unknown scalar field, $\dot{u} = \partial u / \partial t$, $\Omega \in \mathbb{R}^2$ is an open, bounded domain with a Lipschitz continuous boundary $\Gamma = \partial\Omega = \Gamma_D \cup \Gamma_N$, I is a time domain, and ρ, c_p, k, β, f, u_D, u_N, and u_0 are suitable problem data defined on space-time domain and H^1 is a standard Sobolev space.

2.2 Discretization in time and space

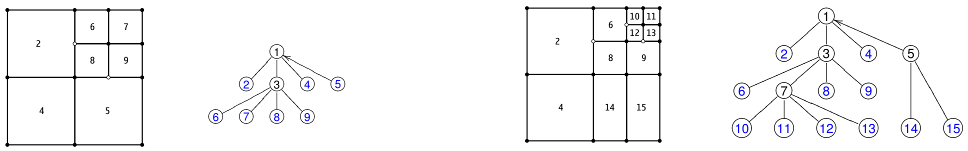
FE-discretization in space gives the following matrix system: $\mathbf{M}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}$ where $\mathbf{M}$ is the mass matrix, $\mathbf{K}$ is the stiffness matrix, and $\mathbf{f}$ is a load vector. Applying the trapezoidal rule for the time discretization we obtain the following algebraic problem $(\mathbf{M} + \alpha\,\delta\mathbf{K})\,\mathbf{u}^{k+1} = [\mathbf{M} - (1-\alpha)\,\delta\mathbf{K}]\,\mathbf{u}^k + \delta\mathbf{f}^k$ where $\mathbf{u}^k$ is a known solution from the previous time step (and $\mathbf{u}^0$ is given), $\mathbf{u}^{k+1}$ is an unknown solution from the current time step, δ is a time step, and parameter $\alpha \in [0, 1]$ that gives different time integration schemes, e.g., for $\alpha = 0$ we obtain explicit Euler scheme, and for $\alpha = 0.5$ — Crank-Nicolson scheme, which we use hereafter.

3 Automatic hp-adaptive 2D code for parabolic problems

The non-stationary self-adaptive *hp*-FEM code presented utilizes identical automatic *hp* adaptive algorithm for every time step, like the stationary code presented in [6]. In particular, the code utilizes quadrilateral finite elements with hierarchic shape functions (integrated Legendre polynomials). It allows for variable order of interpolation over finite elements, preserving C^0 inter-element continuity. The *hp*-FEM code supports 1-irregularity rule with hanging nodes. In other words, an edge can be broken only once without breaking adjacent large elements.

4 Strategy for dealing with incompatible grids

The finite elements are stored in a tree-like structure, in order to allow for convenient mesh refinements. The following rules apply: (1) Actual mesh is represented as leaves of the refinement tree. (2) The tree describes unambiguously a sequence of refinements. (3) Succeeding meshes from the sequence of refined grids are nested. The exemplary sequence of mesh refinements and the corresponding refinement trees are presented in Figure 1b.



(a) mesh + refinement tree, level X (b) mesh + refinement tree, level X+1

In the non-stationary version of self-adaptive hp-FEM, there is a need for storing the old solution $\mathbf{u}^k$, generation of the new mesh, and transferring the old solution onto a new mesh. This is not an easy task, since $\mathbf{u}^k$ must be expressed in terms of degrees-of-freedom of a FE on the new mesh. In general case, the old and the new mesh may be incompatible (in a sense of refinement tree) due to the evolution of solution u in the time; compare two grids and the corresponding refinement trees in Figure 2.

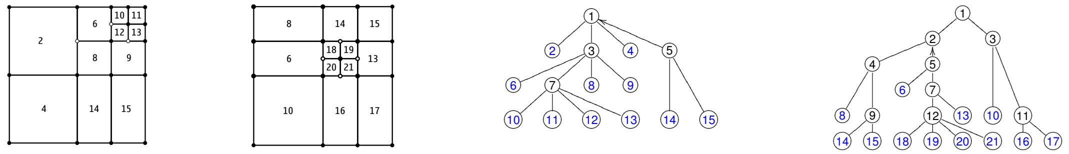
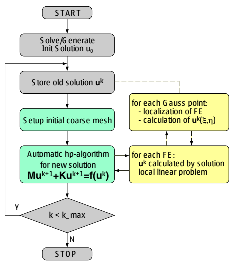


Figure 2: (a) mesh from time step $\mathbf{u}^k$; (b) mesh from time step $\mathbf{u}^{k+1}$; (c) refinement tree - step $\mathbf{u}^k$; (d) refinement tree - step $\mathbf{u}^{k+1}$

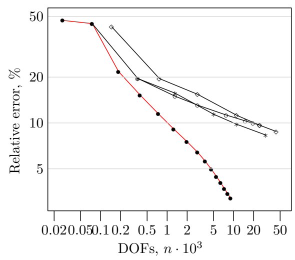
There are two possible strategies for dealing with refinements in the consecutive time steps. The first strategy assumes that the new solution is obtained by the execution of the hp-refinement strategy starting from the very coarse mesh, at every time step. The advantage of this approach is that the refinement tree can freely adapt to the new solution. On the other hand, full process of the hp-adaptation is needed for one time step, and the solution from the previous time step must be projected onto the new mesh in every iteration of the self-adaptive hp-FEM for a new time step. The second strategy assumes that the new solution is obtained by execution of the hp-refinement strategy starting from the previous time step mesh, unrefined in some way. In this case, only last few steps of the the hp-adaptation procedure are needed for one time step. On the other hand, a special strategy of mesh unrefinement is needed, which can also result in possible lost of accuracy of the old solution.

In this paper we select the first strategy, since it is easier to implement. The resulting self-adaptive hp-FEM algorithm for non-stationary problems is presented in Figure 3a

5 Numerical experiments

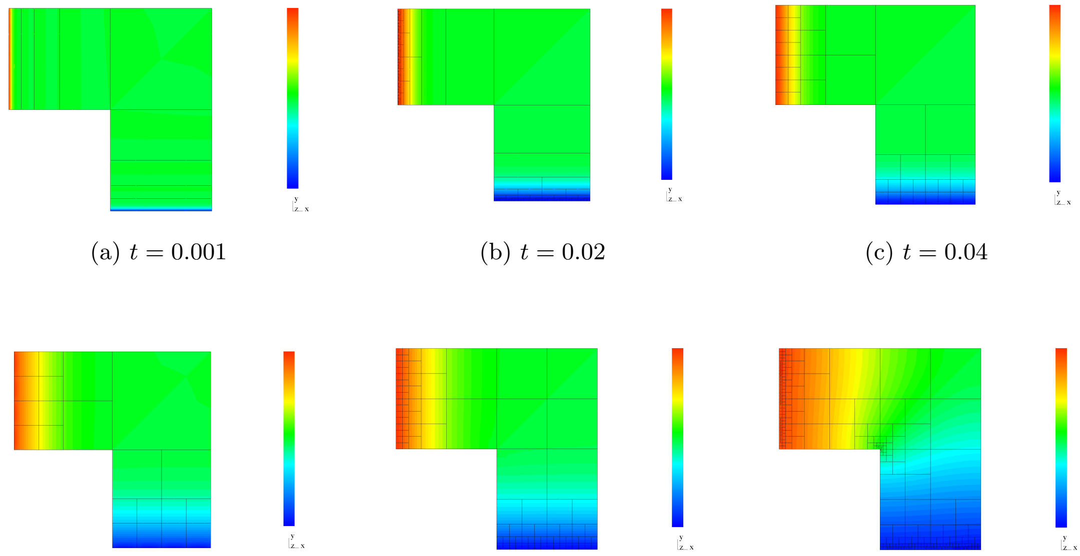


(a) The fully automatic hp adaptive finite element method algorithm for non-stationary problems



(b) Norm of error vs. number DOFs

In this simulation we select the initial temperature distribution u_0 at $t = 0$. The convective heating and cooling starts at $t = 0$ with $k = 1$, $\rho = 1$, $c_p = 1$, $\beta = 1$, $u_N = 1$ (top left edge), and $u_N = -1$ (bottom edge). There are no internal heat sources $f = 0$ and flux through the boundary $q = 0$. The self-adaptive *hp*-FEM algorithm executed at every time step performs *hp* adaptations until the relative error is less than 5%. The temperature distributions are presented in Figure 4. For the numerical solution of the computational problems over *hp*-refined grids in particular time steps we utilize the multi-frontal direct solver [8, 9] with MUMPS package [1, 2, 3]. The self-adaptive *hp*-FEM delivers an exponential computational cost at every time step of the non-stationary problem, as presented in Figure 3b. The *hp*-FEM strategy is compared against the uniform h adaptation with uniform polynomial order of approximation $p = 1$ ($\circ$), $p = 2$ ($*$), and $p = 3$ ($\diamond$) in Figure 3b.



(a) $t = 0.001$ (b) $t = 0.02$ (c) $t = 0.04$

(d) $t = 0.1$ (e) $t = 0.15$ (f) $t = 1.5$

Figure 4: Temperature distribution

6 Conclusion

The automatic *hp*-adaptive FEM code was extended to solve the non-stationary heat transfer problem. The *hp*-refinement iterations were started from the initial coarse mesh for every new time step. For each new time step, the previous time step solution with its optimal mesh was stored. The previous step solution was projected into a sequence of meshes generated in current iteration and utilized to compute the next time step. In our simulation we have not encountered a deadlock problem as reported in [20].

References

[1] Amestoy P.R. and Duff, I.S.: Multifrontal parallel distributed symmetric and unsymmetric solvers, *Computer Methods in Applied Mechanics and Engineering*, (2000) 184, 501-520.

[2] Amestoy, P. R. and Duff, I. S. and Koster, J. and L'Excellent, J.Y.: A fully asynchronous multifrontal solver using distributed dynamic scheduling, *SIAM Journal of Matrix Analysis and Applications*, (2001) 1(23), 15-41.

[3] Amestoy, P. R. and Guermouche, A. and L'Excellent, J.-Y. and Pralet, S.: Hybrid scheduling for the parallel solution of linear systems, *Computer Methods in Applied Mechanics and Engineering*, (2001) 2(32), 136-156.

[4] I. Babuška, B. Guo: The *hp*-version of the finite element method, Part I: The basic approximation results, *Computational Mechanics*, 1 (1986) 21-41.

[5] I. Babuška, B. Guo B.: The *hp*-version of the finite element method, Part II: General results and applications. *Computational. Mechanics*, 1 (1986) 203-220.

[6] Demkowicz, L.: *Computing with hp-Adaptive Finite Elements, Vol. I. One and Two Dimensional Elliptic and Maxwell Problems*, Chapman and Hall/Crc Applied Mathematics and Nonlinear Science (2006)

[7] Demkowicz, L. and Kurtz, J. and D. and Paszyński, M. and Rachowicz, M. and Zdunek, A.: *Computing with hp-Adaptive Finite Elements, Vol. II. Frontiers: Three Dimensional Elliptic and Maxwell Problems with Applications*, Chapman and Hall/Crc Applied Mathematics and Nonlinear Science (2007)

[8] Duff, I. S. and Reid, J. K.: The multifrontal solution of indefinite sparse symmetric linear systems, *ACM Transactions on Mathematical Software* (1983) 9, 302-325.

[9] Duff, I. S. and Reid, J. K.: The multifrontal solution of unsymmetric sets of linear systems, *SIAM Journal on Scientific and Statistical Computing*, (1984) 5, 633-641.

[10] Matuszyk, P. and Paszyński, M.: A Fully Automatic hp Finite Element Method for Stokes Problem in Two Dimensions, *Computer Methods in Applied Mechanics and Engineering* (2008) 197(51-52), 4549-4558.

[11] Paszyński, M. and Pardo, D. Paszyńska: A. Parallel multi-frontal solver for p adaptive finite element modeling of multi-physics computational problems, *Journal of Computational Science* (2010) 1(1) (2010) 48-54.

[12] Paszynska, A. and Paszynski, M. and Grabska,E.: Graph Transformations for Modeling hp-Adaptive Finite Element Method with Mixed Triangular and Rectangular Elements, *Lecture Notes in Computer Science*, (2009) 5545, 875-884.

[13] Paszynska, A. and Paszynski, M. and Grabska, A.: Graph transformations for modeling hp-adaptive Finite Element Method with triangular elements, *Lecture Notes in Computer Science* (2008) 5103, 604-613.

[14] Paszynska A., Grabska E., Paszynski M., A Graph Grammar Model of the hp Adaptive Three Dimensional Finite Element Method. Part I, *Fundamenta Informaticae*, (2012) 114(2) 149-182.

[15] Paszynska A., Grabska E., Paszynski M., A Graph Grammar Model of the hp Adaptive Three Dimensional Finite Element Method. Part II, *Fundamenta Informaticae*, (2012) 114(2) 183-201.

[16] Paszynski, M.: On the Parallelization of Self-Adaptive hp-Finite Element Methods Part I. Composite Programmable Graph GrammarModel, *Fundamenta Informaticae* (2009) 93(4) 411-434.

[17] Paszynski, M.: On the Parallelization of Self-Adaptive hp-Finite Element Methods Part II. Partitioning Communication Agglomeration Mapping Analysis, *Fundamenta Informaticae* (2009) 93(4) 435-457.

[18] M. Paszynski, D. Pardo, A. Paszynska, L. Demkowicz, Out-of-core multi-frontal solver for multi-physics hp adaptive problems, Procedia Computer Science, 4 (2011) 1788-1797

[19] Paszynski, M. and Schaefer, R.: Graph grammar-driven parallel partial differential equation solver, *Concurrency and Computation: Practice and Experience*, (2010) 22(9) 1063-1097,

[20] A. Szymczak, A. Paszyńska, M. Paszyński, D. Pardo: Preventing deadlock during anisotropic 2D mesh adaptation in hp-adaptive FEM, *Journal of Computational Science* (2013)